**K007自动化部署文档**

目录

[1概况 3](#_Toc28191955)

[2注意事项 3](#_Toc28191956)

[3 变量修改 4](#_Toc28191957)

[3.1 Ansible变量修改 4](#_Toc28191958)

[3.2 shell脚本变量修改 12](#_Toc28191959)

[4系统初始化 12](#_Toc28191960)

[4.1 安装ansible 12](#_Toc28191961)

[4.2 免密 12](#_Toc28191962)

[4.3 搭建简易yum源 13](#_Toc28191963)

[4.4 分发repo文件 14](#_Toc28191964)

[4.5 修改主机名 14](#_Toc28191965)

[4.6 配置hosts文件 14](#_Toc28191966)

[4.7 修改root密码 14](#_Toc28191967)

[4.8 创建普通用户 15](#_Toc28191968)

[4.9 系统优化 15](#_Toc28191969)

[4.10 做raid 15](#_Toc28191970)

[4.11 挂盘 15](#_Toc28191971)

[4.12 时间同步 15](#_Toc28191972)

[5 服务部署 16](#_Toc28191973)

[5.1 安装jdk 16](#_Toc28191974)

[5.2 安装node(npm) 16](#_Toc28191975)

[5.3 部署ceph 16](#_Toc28191976)

[5.4 安装zookeeper 16](#_Toc28191977)

[5.5 安装es 17](#_Toc28191978)

[5.6 安装es\_query 17](#_Toc28191979)

[5.7 安装logstash 18](#_Toc28191980)

[5.8 安装kibana 18](#_Toc28191981)

[5.9 安装cerebro 18](#_Toc28191982)

[安装kafka 18](#_Toc28191983)

[5.10安装kafka-manager 19](#_Toc28191984)

[5.11 安装mysql 19](#_Toc28191985)

[5.12 安装neo4j 19](#_Toc28191986)

[5.13 安装postgresql 19](#_Toc28191987)

[5.14 安装nginx 19](#_Toc28191988)

[安装sftp 20](#_Toc28191989)

[5.15 安装redis5 20](#_Toc28191990)

[5.16安装zabbix 20](#_Toc28191991)

[5.16.3 安装zabbix-agent 21](#_Toc28191992)

[6 数据导入 24](#_Toc28191993)

[7 大数据集群部署 24](#_Toc28191994)

[8 业务系统tomcat部署 24](#_Toc28191995)

[9 业务系统web服务部署 27](#_Toc28191996)

## 1概况

本文档是针对K007项目进行的自动化部署方案，优化了现场对基础组件、大数据集群、业务数据、业务系统做出自动化部署流程，从而达到了快速、高效的实施部署效果。部署方式如下：

|  |  |  |
| --- | --- | --- |
| 部署内容 | 部署方式 | 备注 |
| 基础组件 | Ansible | 目前只包含基础组件，业务组件需要手动按照文档进行部署 |
| 大数据集群 | BD-OS部署脚本 | 按需手动添加大数据组件 |
| 业务数据 | Shell执行脚本 |  |
| 业务系统 | Ansible |  |

## 2注意事项

1. 在执行Ansible脚本对系统初始化和部署服务之前要确认网络通畅，保证服务器之间的通信可达。
2. 每执行一个Ansible部署任务的时候请酌情修改相应role下var/main.yml中的变量内容，所有变量不能为空。
3. 中途可能会有如网络原因导致的安装失败或者中断，如果未执行到启动服务的task，则可以重复的执行部署任务，否则要先确认被部署节点的服务是否停止，确认服务已经停止后，方可重复执行。
4. 初次部署需要严格按照文档步骤执行，重复执行，可跳过。
5. 本文档代码git地址为：

**http://git.baifendian.com/DevOps\_yw/K007-ansible.git**

下载代码：git clone <http://git.baifendian.com/DevOps_yw/K007-ansible.git>

## 3 变量修改

### 3.1 Ansible变量修改

Ansible层级如下：

K007-ansible

├── ansible.cfg

├── files

├── hosts

├── manual

├── playbooks

├── README.md

├── roles

└── venv

files：存放业务系统后端部署包

hosts：执行标签

manual：data\_shell----数据导入所用脚本

read-------部署文档存放地址

playbooks：执行文件

roles：执行任务

#### 3.1.1 hosts文件

hosts文件记录所有ansible执行的服务器信息

cd /etc/ansible

vim hosts

根据部署规划，按需更改分组信息

[remote]

[all]

[ssh\_hosts]

[jdk\_hosts]

[node\_hosts]

[sys\_opt\_hosts]

[mysql\_hosts]

[neo4j\_hosts]

[postgresql\_hosts]

[nginx\_hosts]

[host\_name]

[ntp\_hosts]

[zookeeper\_hosts]

[kafka\_hosts]

[kibana\_hosts]

[sftp\_hosts]

[keepalived\_hosts]

[keepalived\_proxy\_hosts]

[logstash\_tse\_hosts]

[kafkamanager\_hosts]

[zabbix\_agent1]

[zabbix\_agent2]

[zabbix\_agent3]

[zabbix\_server]

[zabbix\_proxy]

[node\_export\_hosts]

[cerebro\_hosts]

[redis\_hosts]

[web\_update\_hosts]

[apollo\_hosts]

[system\_hosts]

[securitypolice\_hosts]

[configService\_hosts]

[portal\_hosts]

[personControl\_hosts]

[casejoin\_hosts]

[archiveservice\_hosts]

[flowdesign\_hosts]

[ossService\_hosts]

[controlledObject\_hosts]

[dailyreport\_hosts]

[xxlJob\_hosts]

[warehoureService\_hosts]

#### 3.1.2 roles下变量修改

##### 3.1.2.1 redis5变量修改

cd /etc/ansible/roles/redis\_install

1、按需修改var/main.yml文件修改机器IP信息

2、如果想更换服务端口,修改templates/redis7000.conf.j2和templates/redis7001.conf.j2两个配置文件中端口相关的配置,并对应修改task/main.yml中创建目录的名称、启动命令中指定配置文件的目录以及启动脚本中创建集群命令中的端口信息

##### 3.1.2.2 zabbix变量修改

cd /etc/ansible/roles/zabbix-server\_install/files

修改目录下zabbix.repo文件中以下内容，替换现场源的IP端口：

baseurl=http://172.24.0.58:9930/zabbix/3.0/rhel/7/x86\_64/

cd /etc/ansible/roles/zabbix-server\_install/tasks

修改main.yml文件中第32行代码，更改mysql密码：

shell: mysql -uroot –pk007k007 < /etc/zabbix/create\_zabbix\_user.sql

cd /etc/ansible/roles/zabbix-server\_install/vars

按需修改main.yml文件中以下内容，表示时区：

timezone: Asia/Shanghai

cd /etc/ansible/roles/zabbix-proxy\_install/tasks

修改main.yml文件中第9行代码，更改mysql密码：

shell: mysql -uroot –pk007k007 < /etc/zabbix/create\_proxy\_user.sql

##### 3.1.2.3 Apollo变量修改

ip\_db: 172.16.3.95 #数据库的地址

port\_db: 3306 #数据库端口

user\_db: root #数据库用户名

password\_db: baifendian #数据库密码

##### 3.1.2.4 业务系统前端web修改

cd /etc/ansible/roles/web\_update/vars

修改main.yml文件中注释的代码，必改项：

#远端业务系统服务器中部署程序包的目录

base\_path: /opt/k007-web

#现场替换任务管理域名使用

new\_task\_dns: da-task.cisp.com

#现场域名替换使用

new\_dns: da.cisp.com

#现场替换二楼大屏控制端域名使用

new\_screencontrol\_sec\_dns: da-screen-control-sec.cisp.com

#现场替换一楼大屏控制端域名使用

new\_screencontrol\_dns: da-screen-control.cisp.com

#现场替换一楼大屏系统端域名使用

new\_screenmanager\_dns: da-screen-system.cisp.com

#现场替换一楼大屏展示端域名使用

new\_screen\_web\_dns: da-screen-web.cisp.com

#现场替换二楼大屏展示端域名使用

new\_screen\_web\_sec\_dns: da-screen-web-sec.cisp.com

#现场替换综合研判平台域名使用

new\_integratedpolice\_dns: da-integratedpolice.cisp.com

其他变量按照注释按需修改

##### 3.1.2.5 业务系统后端tomcat参数修改

###### 3.1.2.5.1 系统管理

cd /etc/ansible/roles/system\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.2 基础服务

cd /etc/ansible/roles/securitypolice\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.3 动态本体

cd /etc/ansible/roles/configService\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.4 集成门户

cd /etc/ansible/roles/portal\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.5 任务管理

cd /etc/ansible/roles/projectManager\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.6 情报分析

cd /etc/ansible/roles/information\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.7 专题分析

cd /etc/ansible/roles/bi\_install/vars

修改以下参数：

app\_isRun:sysServer: #系统管理服务的地址 cookie\_key:#biz.propertiestomcatTmpPath: # tomcat临时文件目录，该目录在启动后不能删除，不能是系统/tmp目录下，需要有创建权限basepath: #本地缓存目录export\_path: #导出脚本所在位置web\_path: #前端域名service\_path: #后端域名#postgresql.propertiesjdbc\_url: jdbc:postgresql://192.168.94.35:5432/bi-tenantjdbc\_user: postgresjdbc\_pass: baifendian#quartz.propertiesqzDS\_url: jdbc:postgresql://192.168.94.35:5432/bi-quartz#storage.propertiesjdbc\_address: 192.168.94.35:5432jdbc\_dbName: bi-storage

###### 3.1.2.5.8 犯罪研判平台

cd /etc/ansible/roles/criminalplatform\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.9 比对分析

cd /etc/ansible/roles/compservice\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.10 情报发布

cd /etc/ansible/roles/infopublish\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.11 情报识别

cd /etc/ansible/roles/recognition\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.12 网络情报分析

cd /etc/ansible/roles/networkInfo\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.13 重大事件预警

cd /etc/ansible/roles/eventwarning\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.14 重点人员管控

cd /etc/ansible/roles/personControl\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.15 串并案

cd /etc/ansible/roles/casejoin\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.16 电子档案

cd /etc/ansible/roles/archiveservice\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.17 流程设计中心

cd /etc/ansible/roles/flowdesign\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.18 媒体服务

cd /etc/ansible/roles/ossService\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.19 受控对象

cd /etc/ansible/roles/controlledObject\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.20 安全日报

cd /etc/ansible/roles/dailyreport\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.21 调度服务

cd /etc/ansible/roles/xxlJob\_install/vars

修改以下参数：

apollo\_path:

###### 3.1.2.5.22 数仓接口服务

cd /etc/ansible/roles/warehoureService\_install/vars

修改以下参数：

apollo\_path:

### 3.2 shell脚本变量修改

mysql-recovery-creatdatabase.sh中修改如下参数：

user="root"

password="baifendian"

执行前要保证存在要恢复数据的列表list.txt文件，并确认内容

ceph\_import.sh

执行前要保证存在要恢复数据的列表bucket\_list.txt文件，并确认内容

## 4系统初始化

在部署服务之前需要对服务器进行针对性的优化，保证服务器发挥最大的性能。系统初始化过程包括yum源的搭建，内核参数、网络参数的优化，时间同步，免密等。具体操作步骤如下。

### 4.1 安装ansible

选取一个节点作为ansible-server节点，将上一步骤中用到的repo文件发送到此节点的 /etc/repos.d/ 目录下，然后按照如下步骤安装ansible

在选定的主机安装 ansible 安装命令如下：

yum clean all

yum -y install ansible

rm -rf /etc/ansible

#远程主机发送到ansible-server所在主机

scp ansible.tar.gz [ansible-server\_IP]:/etc/

ssh root@[ansible-server\_IP]]

cd /etc/

tar -zxf ansible.tar.gz

rm -rf ansible.tar.gz

### 4.2 免密

注意：此免密过程针对root用户。

编辑 /etc/ansible/hosts 文件修改 [ssh\_hosts] 标签下的内容，然后执行如下命令安装(标签下要正确的配置每台服务器的ssh端口号和root密码)

ansible-playbook /etc/ansible/playbooks/ssh.yml

验证sshd是否启动

ansible all -m service -a "name=sshd state=started"

如果上面的命令可以执行成功则代表免密完成。

### 4.3 搭建简易yum源（手动调整）

首先挑选一台适合作为yum源的服务器，将一块300G以上的物理磁盘挂/data1目录下，随后上传yum.tar.gz 到/data1/目录下，执行解压命令如下：

cd /data1

tar -zxf yum.tar.gz

python启动临时http代理充当yum源代理，监听9930端口

cd /data1/yum

python -m SimpleHTTPServer 9930 &>/dev/null &

执行如下命令替换repo文件中的IP

find ./yum.repos.d -type f -name \*.repo | xargs perl -pi -e 's|10.4.41.110|172.24.9.58|g'

本机配置yum源repo文件

rm -rf /etc/yum.repos.d/\*

cd /data1/yum

cp yum.repos.d/\* yum.repos.d/

yum clean all

验证是否搭建成功

yum makecache

yum -y install vim

#### 4.3.1 替换yum源代理方式(apache代理)

由于上一步骤中python临时启用的简易yum源不稳定，有并发性能问题，且不好管理，所以利用我在上一步骤中搭建的临时yum源，安装apache服务，重新代理yum源。

yum -y install httpd

#配置文件已经改好，9930端口，代理到/data1/yum目录

cd /data1/yum

cp httpd.conf /etc/httpd/conf/

ps aux | grep python | grep -v grep | awk '{print $2}' | xargs kill -9

systemctl restart httpd

systemctl status httpd

yum clean all

#验证是否搭建成功

yum makecache

yum -y install telnet

### 4.4 分发repo文件

编辑 /etc/ansible/hosts 文件修改 [remote] 标签下的内容，[remote] 标签包含除了ansible(第一个)节点之外的所有节点

#删除所有远端节点的 /etc/yum.repos.d/ 下的文件

ansible remote -m shell -a "rm -rf /etc/yum.repos.d/\*"

#将本机节点 /etc/yum.repos.d/ 下的文件拷贝到远端

ansible remote -m copy -a "src=/etc/yum.repos.d dest=/etc/ owner=root group=root mode=0644"

#7个repo文件

ansible all -m shell -a "ls /etc/yum.repos.d/ | wc -l"

#验证是否搭建成功

ansible all -m shell -a "yum clean all"

#执行这个命令的时候会有点慢

ansible all -m shell -a "yum makecache"

### 4.5 修改主机名

编辑 /etc/ansible/hosts 文件修改 [host\_name] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/host\_name.yml

验证是否修改成功

ansible all -m shell -a "hostname"

### 4.6 配置hosts文件

注意：playbook的yml文件中指定的hosts为all，并不是在 /etc/ansible/hosts 中取得的变量

如果有其他的域名映射关系需要配置在 /etc/hosts 中，请编辑templates/domain\_hosts 文件，在其中添加想要配置的映射关系，每个映射关系一行，映射关系会原封不动的追加到每个节点的 /etc/hosts 文件末尾。

执行如下命令自动生成hosts文件

ansible-playbook /etc/ansible/playbooks/hosts.yml

验证是否配置成功

vim /etc/hosts

### 4.7 修改root密码

执行如下命令即可, 其中 k007 替换成想要修改成的root密码。

ansible all -m shell -a "echo 'k007' | passwd --stdin root"

### 4.8 创建普通用户

下面命令会在每台服务器上创建k007用户，密码为 k007

ansible all -m shell -a "useradd k007 && echo 'k007' | passwd --stdin k007"

### 4.9 系统优化

注意脚本执行成功后需要看swap是否是关闭状态，若不是关闭状态请执行:

swapoff -a

编辑 /etc/ansible/hosts 文件修改 [sys\_opt\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/sys\_opt.yml

验证是否优化完成

上面执行过程未报错，并且下面命令的输出结果为655350，则优化成功

ansible all -m shell -a "ulimit -n"

### 4.10 做raid

编辑hosts文件

### 4.11 挂盘

编辑 /etc/ansible/hosts 文件的内容，然后执行如下

/etc/ansible/hosts 文件中的标签采用动态传入的方式，可以动态的根据自己的需求任意定义标签名称，如下命令中的 hosts=one\_disk\_mount 表示 /etc/ansible/hosts 文件中的 [one\_disk\_mount] 标签，

mount\_sh=ck\_mount.sh表示调用的脚本是 /etc/ansible/roles/auto\_mount/files 目录下的 ck\_mount.sh 脚本（针对不同集群挂载不同挂盘数量的情况，所执行的挂载脚本不同）

ansible-playbook /etc/ansible/playbooks/auto\_mount.yml --extra-vars "hosts=one\_disk\_mount mount\_sh=ck\_mount.sh"

验证是否挂载成功

ansible one\_disk\_mount -m shell -a "lsblk"

### 4.12 时间同步

此任务会在每台服务器节点上安装ntp服务，并自动和配置的ntp-server节点保持时间同步，自动添加定时任务(每分钟同步一次时间)，此任务也会自动的修改时区，(注意：如果时间服务器更改，需要修改vars/main.yml里的变量配置)

编辑 /etc/ansible/hosts 文件修改 [ntp\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/ntp\_install.yml

验证是否成功

crontab -l

/usr/sbin/ntpdate -u 172.24.9.45 ; /usr/sbin/hwclock -w

## 5 服务部署

模块化的方式部署相应的服务组件，每个组件对应一个task任务，部署前要确认hsots文件中的配置是否正确，并仔细检查相应role下var/main.yml中的变量内容，所有变量不能为空。

每个服务部署过程中的最后一步都有自检过程，手动自检方法也单独附加到每个组件部署的末尾，确保检测过程的可控性。

### 5.1 安装jdk

编辑 /etc/ansible/hosts 文件修改 [jdk\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/jdk.yml

验证安装成功

ansible jdk\_hosts -m shell -a "java -version"

### 5.2 安装node(npm)

编辑 /etc/ansible/hosts 文件修改 [node\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/node.yml

验证安装成功

ansible node\_hosts -m shell -a "node -v"

### 5.3 部署ceph

发送ceph\_install文件夹到 ceph主节点所在主机(不需要挂盘)

scp -r ./ceph\_install [ceph-master-ip]:/opt/

cd /opt/ ceph\_install

详细查看 ceph\_install.sh 中配置的变量是否需要修改然后执行如下命令安装。(进行变量修改时按照实际物理机的磁盘数量)

sh ceph\_install.sh

### 5.4 安装zookeeper（按需）

安装zookeeper时要与ambari-server安装的zookeeper分开

编辑 /etc/ansible/hosts 文件修改 [zookeeper\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/zookeeper.yml

验证是否安装成功

ansible zookeeper\_hosts -m wait\_for -a "port=2181 delay=0 timeout=2"

### 5.5 安装es

a.编辑 /etc/ansible/hosts 文件，修改[es\_servers]标签下的内容，并按需修改node\_master和node\_data对应的值

b.修改/etc/ansible/roles/es/handlers/main.yml中的值，根据注释按需更改：

- name: mkdir Subdirectory

#如果数据目录和日志目录与部署路径不在同一块盘，此处需要更改路径为具体地址

file: dest=/opt/elasticsearch/{{ item }} mode=0755 state=directory owner=es group=es

with\_items:

- logs

- data

b. 修改/etc/ansible/roles/es/vars/main.yml中的值，根据注释按需更改：

#ES集群名称

cluster\_name: test-cluster

#ES部署包名，需要注意的是此包已经是做好软连接后的包，如需更改版本，请做好软连后重新打包，再更改此处名称

es\_package\_file: elasticsearch-5.2.2.tar.gz

#ES具体的包名，与上述名称保持一致即可

es\_package: /opt/elasticsearch-5.2.2

#ES工作路径。基本默认，按需更改

es\_dir: /opt/elasticsearch

#ES配置文件

config\_name: elasticsearch.yml

#ES数据目录地址

path\_data: /opt/elasticsearch/data/

#ES日志地址

path\_logs: /opt/elasticsearch/logs/

#ES连接节点，只写主节点即可，host 或者 host:port

unicast\_hosts: '["172.24.9.40:9300","172.24.9.41:9300","172.24.9.42:9300"]'

#集群允许最多的主节点数量，按需修改

minimum\_master\_nodes: 1

c.安装es，执行下面命令：

ansible-playbook /etc/ansible/playbooks/es-5.yml

### 5.6 安装es\_head

修改/etc/anisble/hosts下的标签信息

[es\_plugin\_head]

172.24.9.42

按需修改var/main.yml中变量信息

head\_package\_file: elasticsearch-head.tar.gz

head\_dir: /opt/elasticsearch-head

执行部署命令：

ansible-playbook /etc/ansible/playbooks/es\_head.yml

### 5.7 安装logstash

编辑 /etc/ansible/hosts 文件，修改相应的内容，然后执行如下

/etc/ansible/hosts 文件中的主机标签采用动态传入的方式，如下命令中的 hosts=logstash\_tse\_hosts 表示 /etc/ansible/hosts 文件中的 [logstash\_tse\_hosts] 标签

Tse、ala、other集群：

ansible-playbook /etc/ansible/playbooks/logstash.yml --extra-vars "hosts=logstash\_tse\_hosts"

ansible-playbook /etc/ansible/playbooks/logstash.yml --extra-vars "hosts=logstash\_ala\_hosts"

ansible-playbook /etc/ansible/playbooks/logstash.yml --extra-vars "hosts=logstash\_other\_hosts"

检测端口是否启动

ansible logstash\_tse\_hosts -m wait\_for -a "port=9600 delay=0 timeout=2"

ansible logstash\_ala\_hosts -m wait\_for -a "port=9600 delay=0 timeout=2"

ansible logstash\_other\_hosts -m wait\_for -a "port=9600 delay=0 timeout=2"

### 5.8 安装kibana

编辑 /etc/ansible/hosts 文件修改 [kibana\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/kibana.yml

检测端口是否启动

ansible kibana\_hosts -m wait\_for -a "port=5601 delay=0 timeout=2"

### 5.9 安装cerebro

编辑 /etc/ansible/hosts 文件修改 [cerebro\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/cerebro.yml

验证

ansible crebro\_hosts -m wait\_for -a "port=1234 delay=0 timeout=2"

### 5.10安装kafka-manager

编辑 /etc/ansible/hosts 文件修改 [kafkamanager\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/playbooks/kafka-manager.yml

验证是否启动成功

ansible kafkamanager\_hosts -m wait\_for -a "port=23333 delay=0 timeout=2"

### 5.11 安装mysql

编辑 /etc/ansible/hosts 文件修改 [mysql\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/mysql.yml

验证是否安装成功

ansible mysql\_hosts -m shell -a "systemctl status mysqld"

或

ansible mysql\_hosts -m service -a "name=mysqld state=started"

### 5.12 安装neo4j

编辑 /etc/ansible/hosts 文件修改 [neo4j\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/neo4j.yml

验证是否安装成功

ansible neo4j\_hosts -m wait\_for -a "port=7474 delay=0 timeout=2"

### 5.13 安装postgresql

编辑 /etc/ansible/hosts 文件修改 [postgresql\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/postgresql.yml

验证是否安装成功

ansible postgresql\_hosts -m wait\_for -a "port=5432 delay=0 timeout=2"

### 5.14 安装nginx

编辑 /etc/ansible/hosts 文件修改 [nginx\_hosts] 标签下的内容，替换nginx所部署的服务器IP地址

[nginx\_hosts]

172.24.9.42 nginx\_hosts=172.24.9.42

var/main.yml文件中按照注释按需修改

#nginx版本

nginx\_version: nginx-1.13.4-1.el7.ngx.x86\_64

#域名前缀，按需修改

nginx\_begin: da

#域名中缀，按需修改

nginx\_mid: -pro

#域名后缀，按需修改

nginx\_end: cisp.com

#前端放置路径

web\_path: /opt/k007-web

#########################################################################

#########################################################################

system\_hosts: 172.24.4.41

archiveservice\_hosts: 172.24.4.41

portal\_hosts: 172.24.4.42

eventwarning\_hosts: 172.24.4.41

recognition\_hosts: 172.24.4.41

information\_hosts: 172.24.4.43

networkInfo\_hosts: 172.24.4.41

casejoin\_hosts: 172.24.4.41

personControl\_hosts: 172.24.4.41

bi\_hosts: 172.24.4.41

infopublish\_hosts: 172.24.4.42

criminalplatform\_hosts: 172.24.4.42

compservice\_hosts: 172.24.4.42

flowdesign\_hosts: 172.24.4.42

projectManager\_hosts: 172.24.4.43

securitypolice\_hosts: 172.24.4.42

ossService\_hosts: 172.24.4.42

warehoureService\_hosts: 172.24.4.43

dailyreport\_hosts: 172.24.4.43

controlledObject\_hosts: 172.24.4.43

xxl\_job\_hosts: 172.24.4.42

然后执行如下

ansible-playbook /etc/ansible/playbooks/nginx.yml

验证是否安装成功(如果不是监听的80端口，请更下下面的port 变量)

ansible nginx\_hosts -m wait\_for -a "port=80 delay=5 timeout=10"

### 5.15 安装redis5

1. 需要修改一下内容： 1.hosts文件[redis\_hosts]下的ip信息 2.var/main.yml文件修改机器IP信息 3.如果想更换服务端口,修改templates/redis7000.conf.j2和templates/redis7001.conf.j2两个配置文件中端口相关的配置,并对应修改task/main.yml中创建目录的名称、启动命令中指定配置文件的目录以及启动脚本中创建集群命令中的端口信息
2. 编辑 /etc/ansible/hosts 文件修改 [redis\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/redis.yml

### 5.16安装zabbix

#### 5.16.1安装node\_export

编辑 /etc/ansible/hosts 文件修改 [node\_export\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/node\_export.yml

验证

ansible node\_export\_hosts -m service -a "name=node\_exporter state=started"

或

ansible node\_export\_hosts -m shell -a "systemctl status node\_exporter"

#### 5.16.2 安装zabbix-server

安装zabbix-server：编辑/etc/ansible/hosts 文件修改 [zabbix\_server] 标签下的内容, 然后执行如下

ansible-playbook /etc/ansible/playbooks/zabbix-server.yml

验证

ansible zabbix\_server -m wait\_for -a "port=10051 delay=0 timeout=2"

#### **5.16.3 安装zabbix-agent**

1. 编辑/etc/ansible/hosts 文件修改 [zabbix\_agent] 标签下的内容

(2)在--extra-vars "zabbix\_server=xxx.xxx.xxx.xxx"中填写对应zabbix-server的IP，然后执行如下

ansible-playbook /etc/ansible/playbooks/zabbix-agent.yml --extra-vars "zabbix\_server=172.24.9.46"

验证：

ansible zabbix\_agent -m wait\_for -a "port=10050 delay=0 timeout=2"

#### 5.16.4 安装zabbix-proxy

1. 编辑/etc/ansible/hosts 文件修改 [zabbix\_proxy] 标签下的内容

(2)在--extra-vars "zabbix\_server=xxx.xxx.xxx.xxx"中填写对应zabbix-server的IP，然后执行如下

ansible-playbook /etc/ansible/playbooks/zabbix-proxy.yml --extra-vars "zabbix\_server=172.24.9.46"

验证：

ansible zabbix\_proxy -m wait\_for -a "port=10051 delay=0 timeout=2"

配置proxy-agent:

1. 编辑/etc/ansible/hosts 文件修改 [proxy\_agent] 标签下的内容

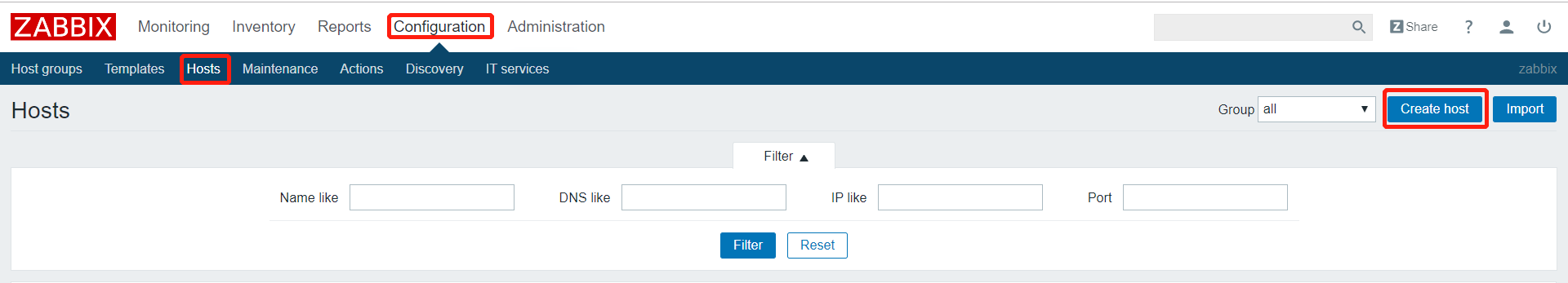
(2)在--extra-vars "zabbix\_server=xxx.xxx.xxx.xxx zabbix\_proxy=xxx.xxx.xxx.xxx"中填写对应zabbix-server和zabbix\_proxy的IP，然后执行如下

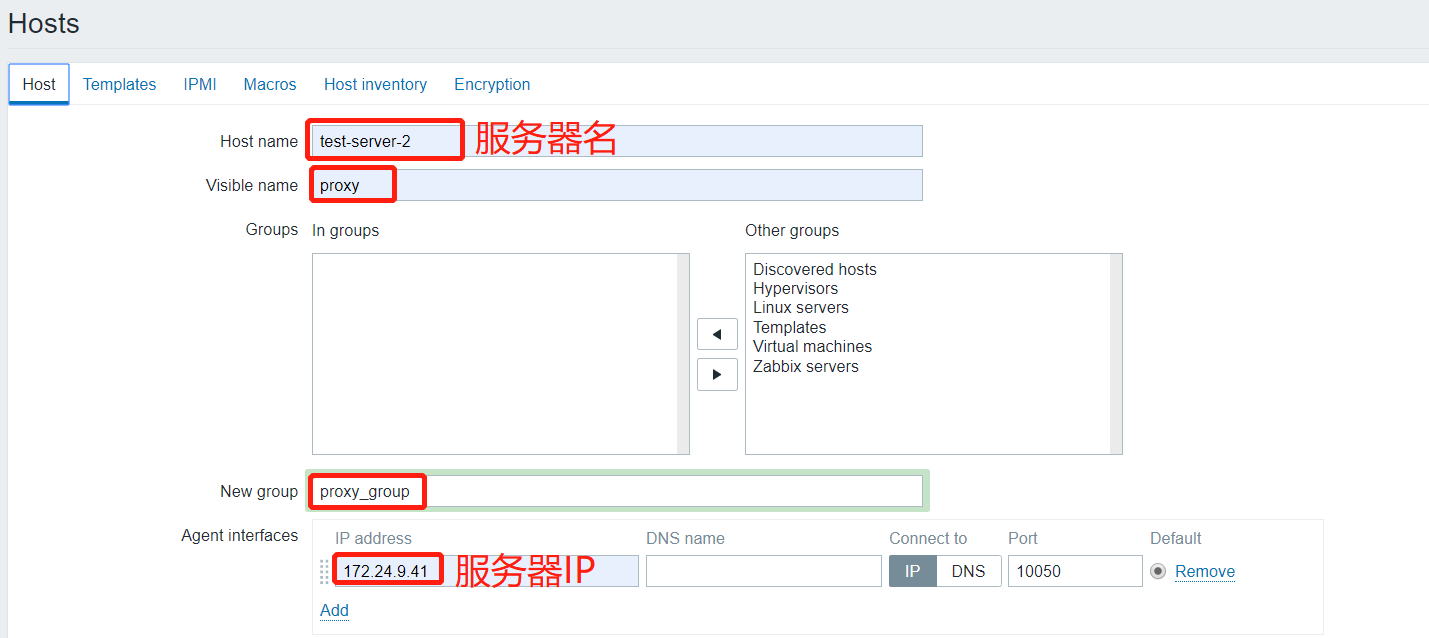
ansible-playbook proxy-agent.yml --extra-vars "zabbix\_server=172.24.9.46 zabbix\_proxy=172.24.9.55"

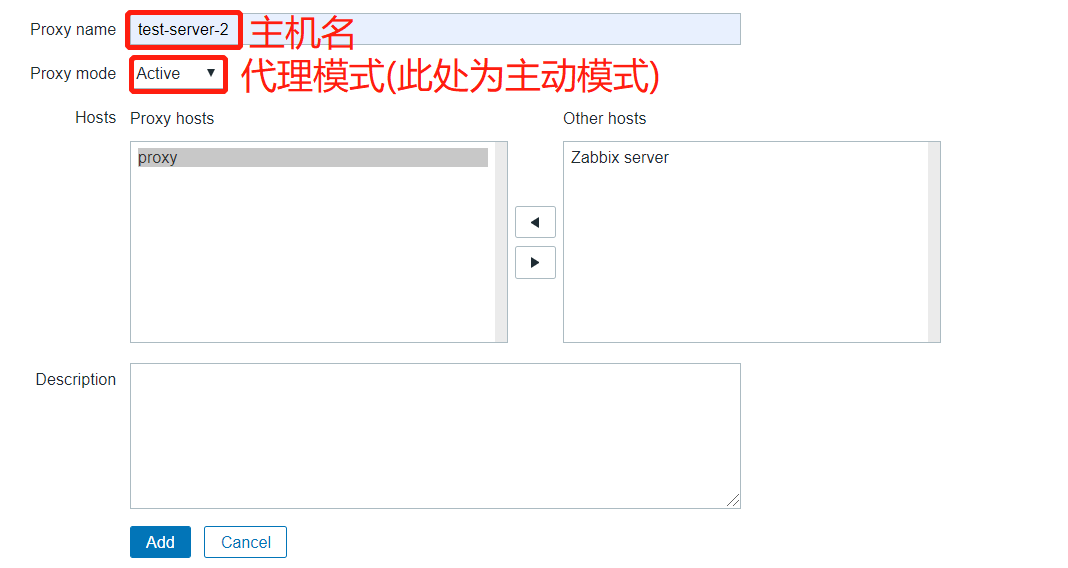
验证：

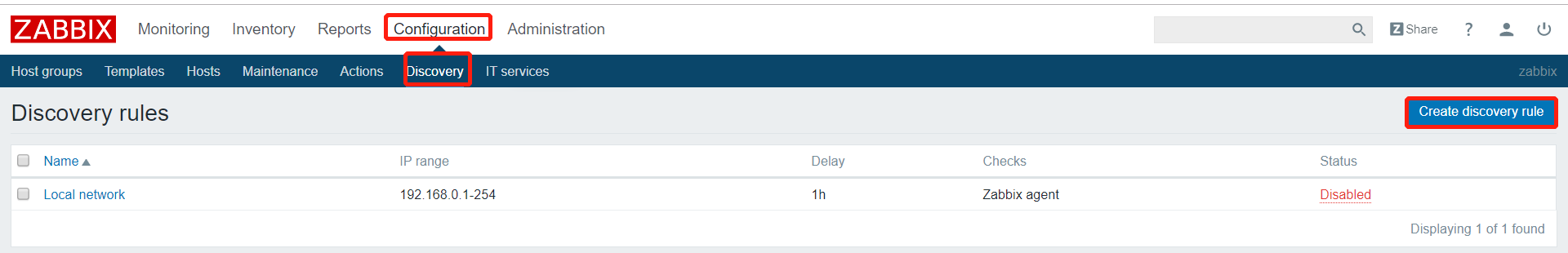
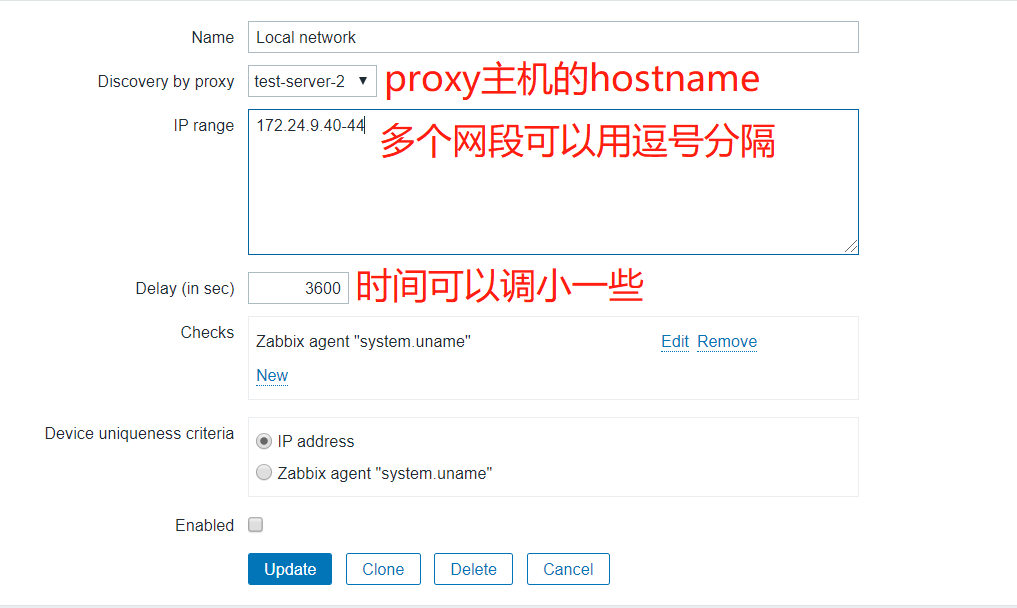
ansible proxy\_agent -m wait\_for -a "port=10050 delay=0 timeout=2"

web页面配置:(有图有真相)

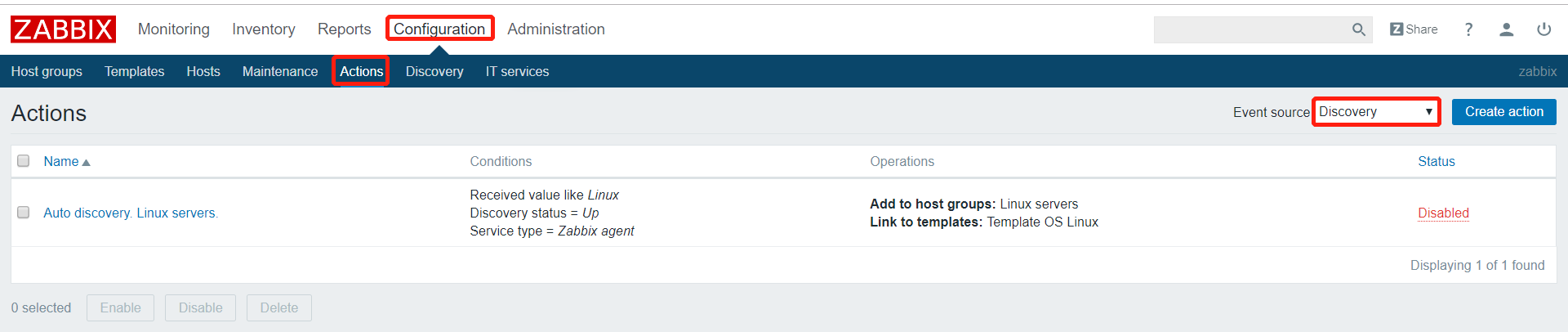
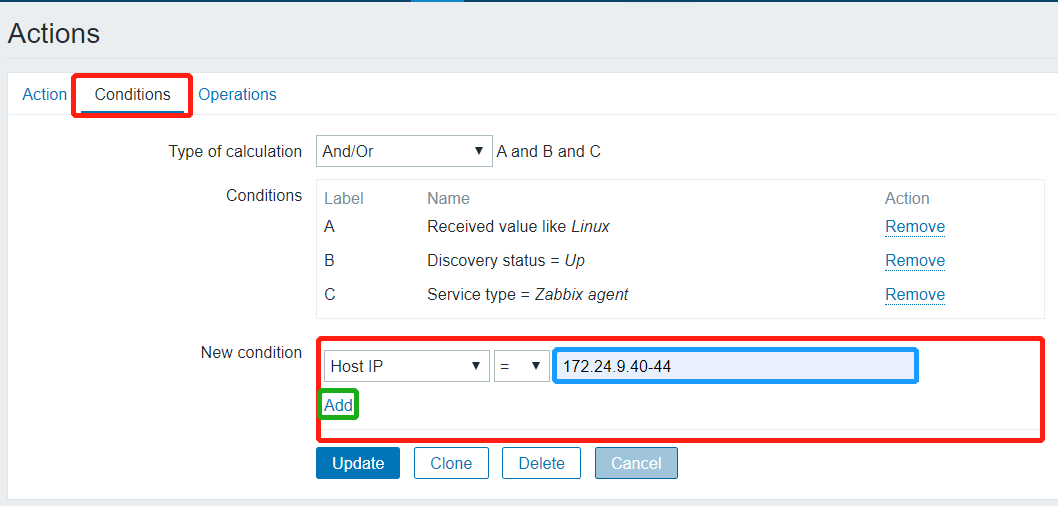
1)创建proxy主机

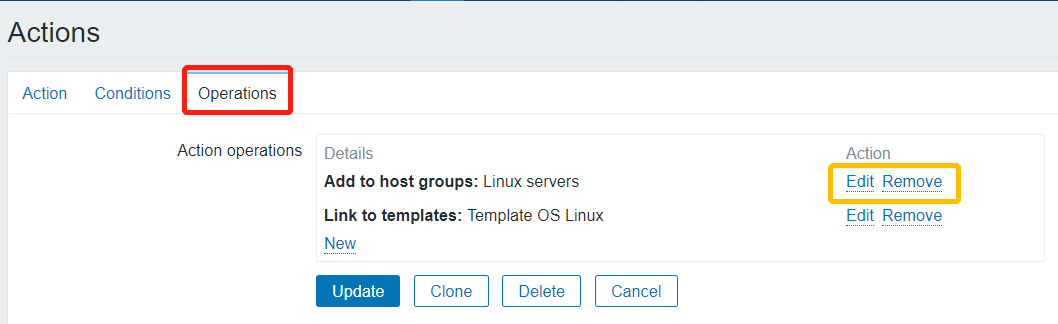


2)创建代理

3)创建自动发现策略 

这时可以看到发现的机器，但是却不能监控

4)配置动作将发现的机器自动添加到Hosts中进行监控 



# 6 数据导入

cd /etc/ansible/manual/data\_shell

1. 将恢复脚本发送到各个数据库所在服务器节点
2. sh xxx.sh命令执行各个脚本针对数据目录进行恢复

# 7 大数据集群部署

安装包说明

| **安装包名称** | **文件类型** | **说明** |
| --- | --- | --- |
| InstallerV3.1 | 目录 | 整个安装包的根目录。 |
| BD-OS | 目录 | 整个BD-OS安装包存放目录 |
| bin | 目录 | Installer的可执行安装脚本存放目录 |
| key | 目录 | 自动化安装时生成的key都存放在些目录下。 |
| log | 目录 | 日志目录。 |
| stacks | 目录 | 所有插件的存放目录索引。 |
| common-services | 目录 | 所有插件的存放目录。 |
| tmp | 目录 | 自动化部署过程中的临时目录。 |
| tool | 目录 | 存放jdk,python,ntp和自动化安装包的目录。 |

## 7.1 Ambari server安装

**1初始化及准备工作**

* 上传BD-OS的部署脚本installerV3.1\_oversea.tar。
* 将安装文件拷贝到任意目录然后解压

**2****系统初始化setup1**

* 进入bin目录下，编辑host.file文件用来做服务器之间免密打通

[root@hlg-5p94-zhangshumei installerV3.1]# cat host.file 172.24.5.92:root:VETaojikj2M= #IP：root用户：root密码，冒号分割

172.24.5.93:root:5PK77h3IPfk=

172.24.5.94:root:6cOO2z3C5os=

* 之后执行完整系统初始化脚本setup1

./setup1 local\_ip ssh\_port

注明： local\_ip是本机的内网ip；ssh\_port是本机的ssh端口号；

该脚本会进行多主机服务器的免密打通、系统环境的检查（checklist）,初始化系统配置参数、安装yum本地源，安装必要的系统依赖包，分发安装jdk，并作时间同步等。

**3软件初始化setup2**

* Setup1正确执行完成后，执行软件安装脚本setup2

**./setup2**

注意：这里生成mysql默认的root密码存储在当前bin目录的./mysql\_passwd中，后续使用到该密码查看该文件即可

Setup2安装成功，操作系统脚本安装就结束了

如果安装过程中出现错误，可以查看日志分析(1.log,2.log)原因，之后可以通过单功能脚本进行分步补充安装，脚本如下几个：

**表 脚本说明**

| **脚本名称** | **执行参考** | **说明** |
| --- | --- | --- |
| sshnopass | sshnopass local\_ip ssh\_port | 自动化打通ambari-server到各agent节点免密功能的脚本。 |
| checklist.sh | sh checklist.sh | 系统环境检查脚本 |
| yum\_install | yum\_install local\_ip | 自动化安装配置yum源脚本 |
| distribute | Distribute local\_ip ssh\_port | 安装部署jdk,python,ntp等软件的脚本。 |
| mysql\_install | mysql\_install install | 自动化安装部署mysql的脚本。 |
| ambari\_install | Ambari\_install install ambari\_passwd | 自动化安装配置ambari的脚本。 |

回退和删除mysql和ambari

如果安装有问题，可以进行单独软件删除操作，之后重新执行安装脚本。

* 卸载msyql 执行

./mysql\_install remove

之后执行rpm -e bfd-mysql-5.5.32-1.x86\_64，再删除/opt/bfd/bfd-mysql安装目录。

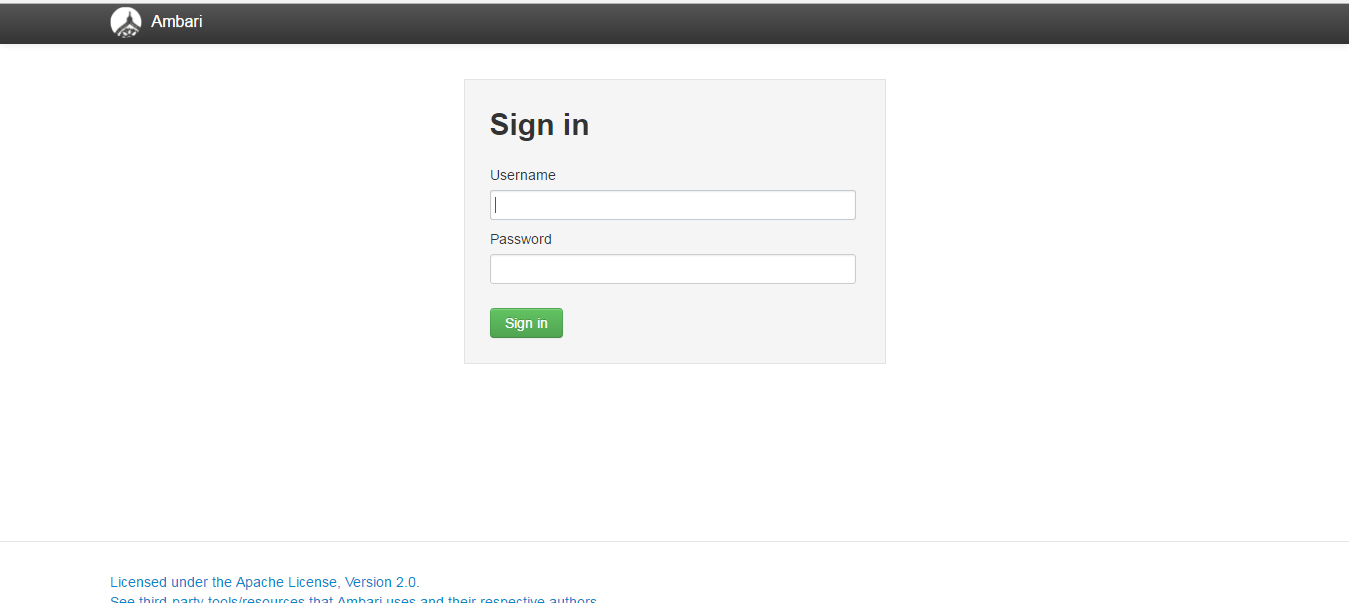
* 如卸载ambari-server执行

./ambari\_install remove

## 7.2使用Ambari部署平台

### 7.2.1登陆ambari

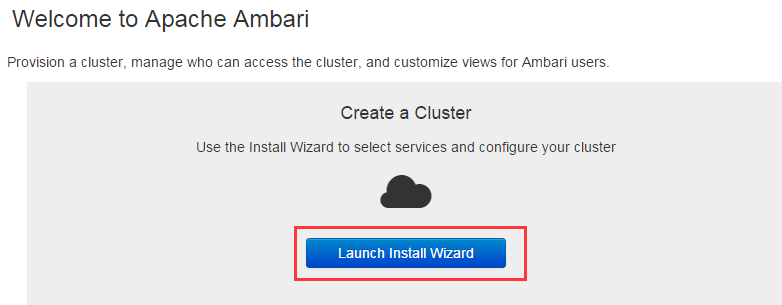
ambari-server启动以后，打开浏览器，访问http://<ambari-serverip>:8080即可进入ambari的登陆界面。



**图 ambari的登陆界面**

默认的用户名和密码为admin admin。

登陆成功后进入如下界面：

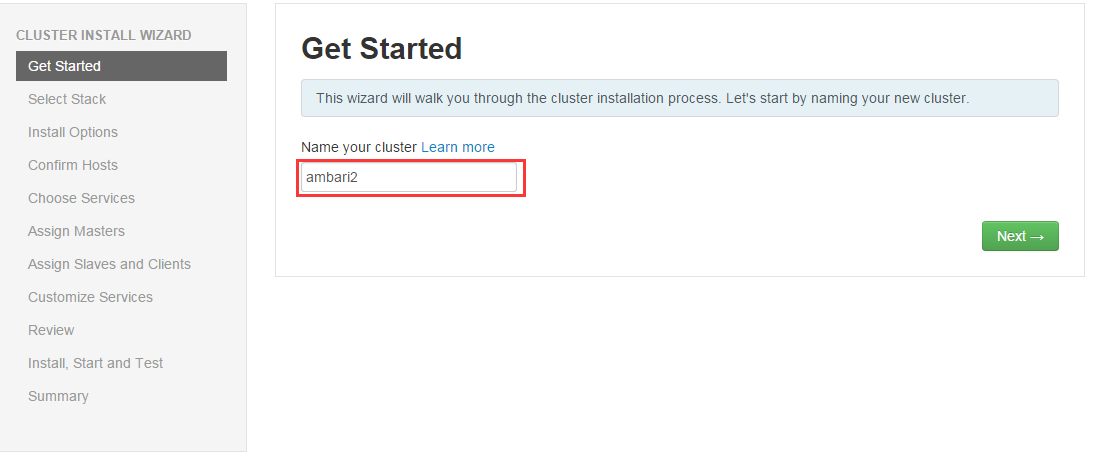


**图 ambari登陆成功界面**

我们点击”Launch Install Wizard”进入安装向导。

### 7.2.2输入集群名称

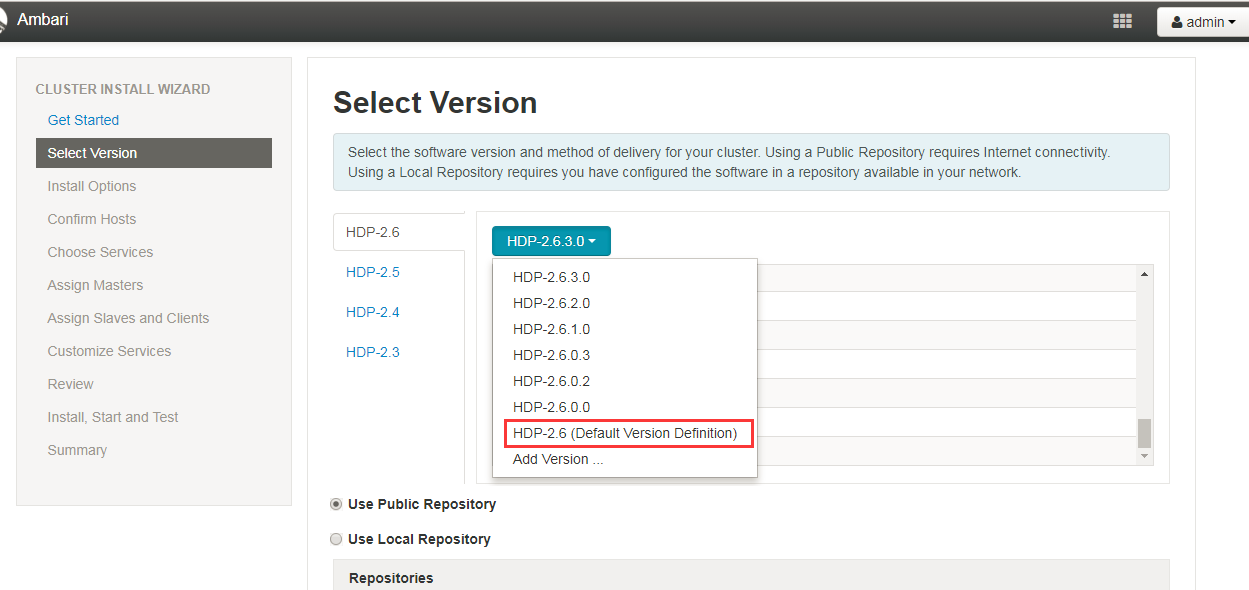
向导第一步，我们要输入集群名称，输入”BigDataCluster”，可根据自身需求自定义。



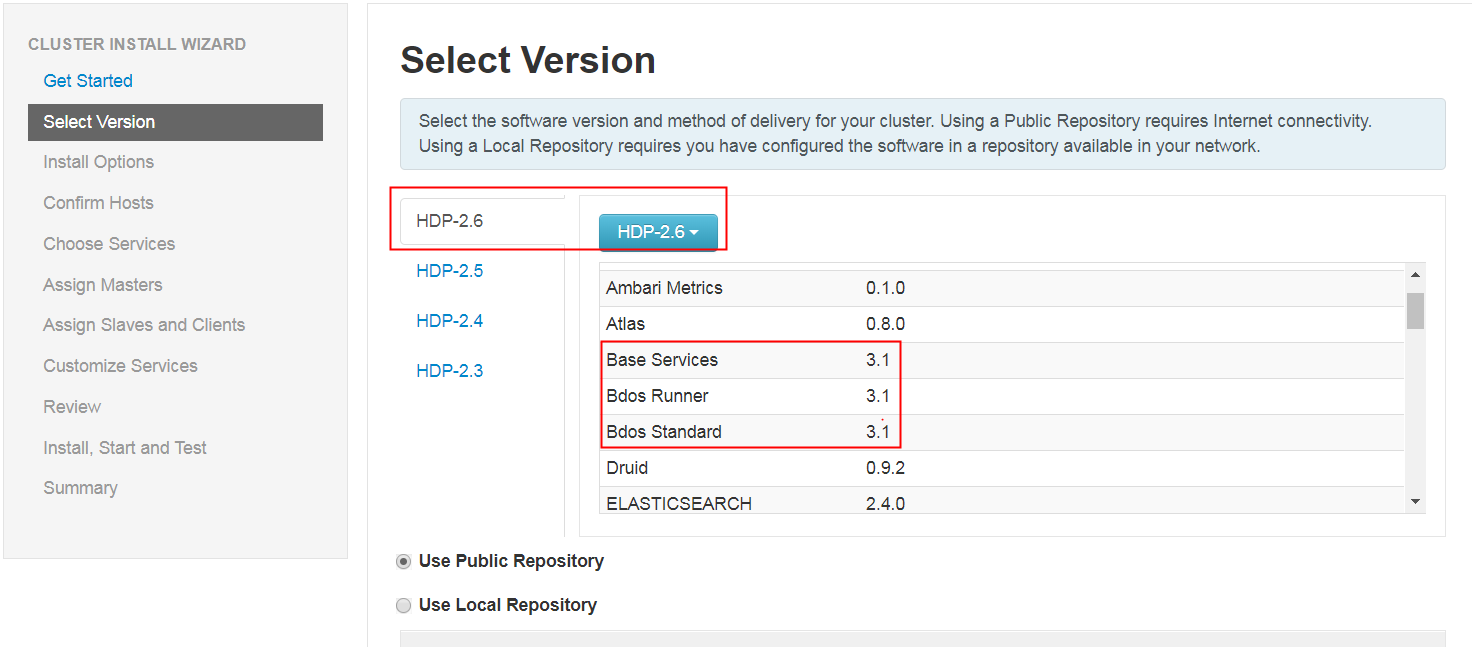
**图 集群名称输入**

### 7.2.3选择要安装的版本

第二步我们选择安装的版本，如下图，我们要选择HDP-2.6.。（第一次没有请F5刷新页面）



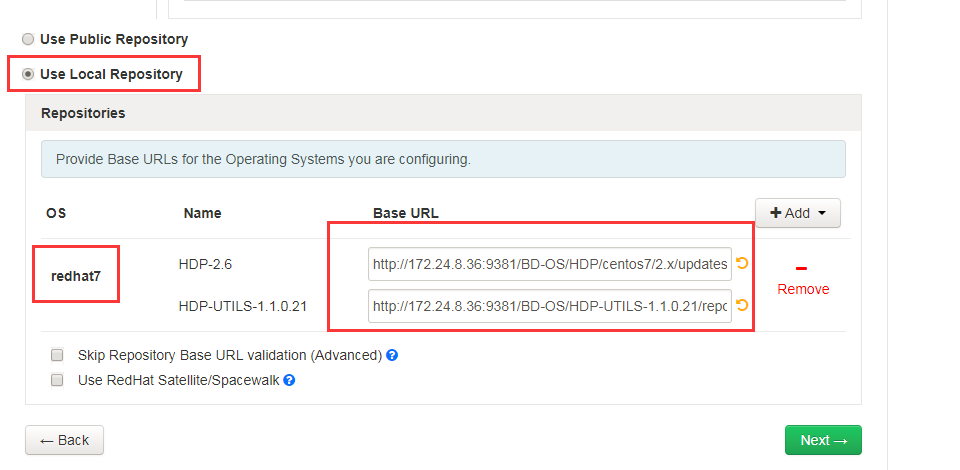
**图 安装版本选择**

****

**图 HDP-2.4**

出现HDP-2.6，选择后可以继续下一步安装

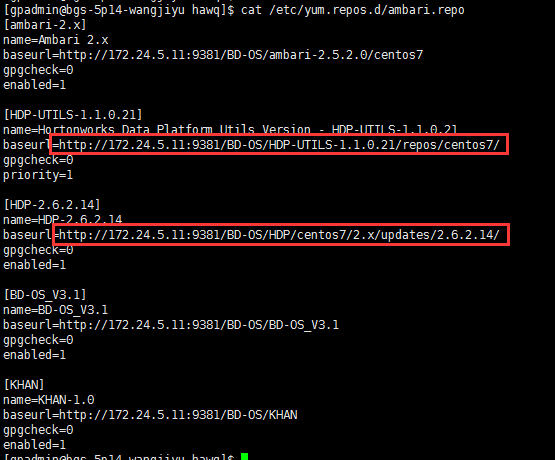
* 选择本地仓库：



**图 选择本地仓库**

这里必须选择为redhat7，HDP-2.6和HDP-UTILS-1.1.0.21的yum源地址来自ambari-server安装机器的/etc/yum.repos.d/ambari.repo文件中。操作如下：

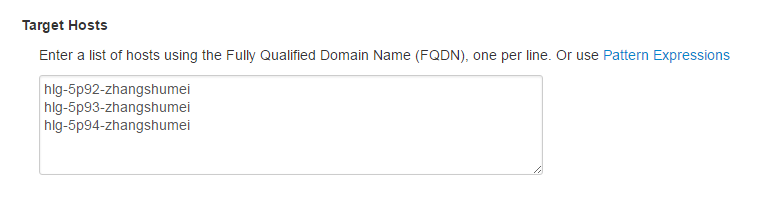
登陆到ambari-server所在的机器，执行cat /etc/yum.repos.d/ambari.repo将下图中圈出来的内容，粘贴到ambari里。



**图 执行程序**

### 7.2.4添加要被ambari管理的主机

输入需要管理的目标主机名称**不能输入ip地址，host名不能大写**，可以使用表达式[]号内部可以数字范围，如下：

hlg-5p9[2-4]-hostname

**图 主机名称**

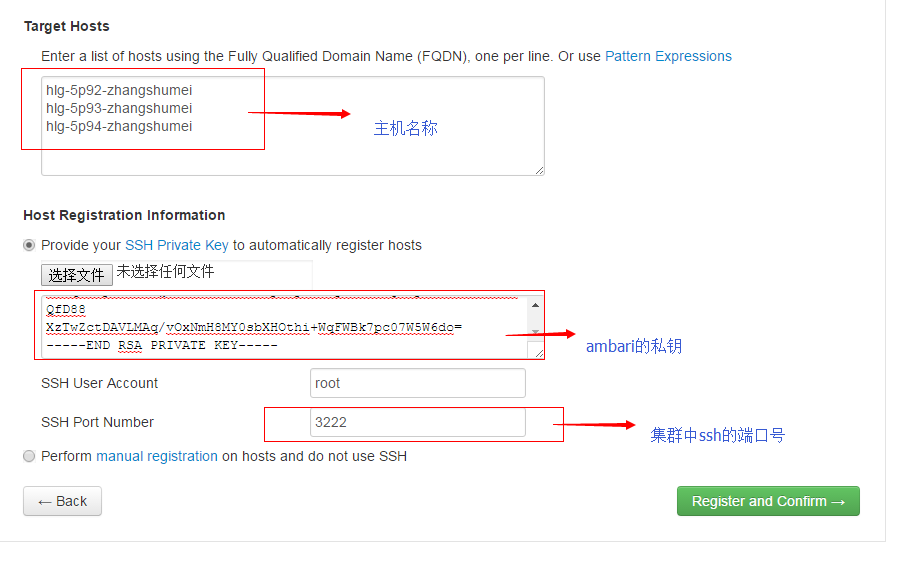
* 输入ssh private key文件

这里需要登陆到ambari-server所在的机器，然后执行：cat /root/.ssh/id\_rsa，将输出的信息粘到如下的输入栏里。



**图 输出信息**

SSH User Account输入root即可，SSH端口要填写机器真实的ssh端口。最终如下图所示：

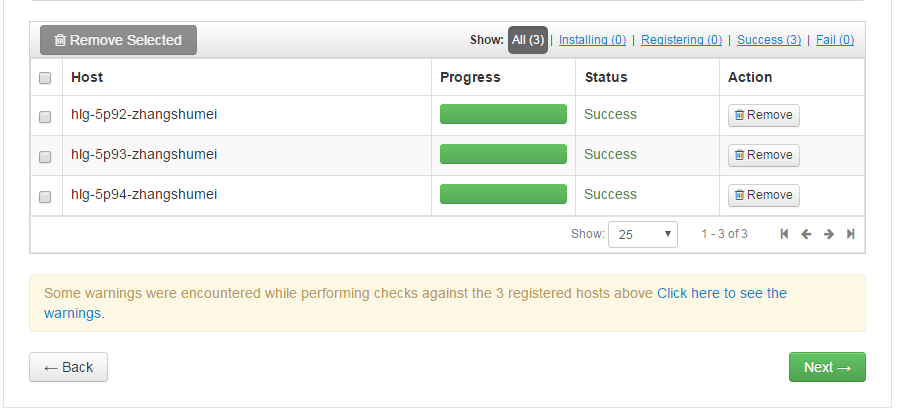


**图 ROOT输入**

### 7.2.5注册主机

点击“Register and Confirm”。

注册主机安装成功以后，点击next开始安装服务。



**图 安装服务**

## 7.3重要说明和注意事项(必看)

1. 服务安装顺序：

因为组件之间有依赖关系，所以请务必按照顺序安装，先安装ZK、HDFS、YARN+Mapreduce组件再安装MYSQL，HIVE，HBASE，SQOOP，SPARK2，REDIS，STORM，KAFKA Ambari Metrics。

1. 服务安装依赖注意事项：

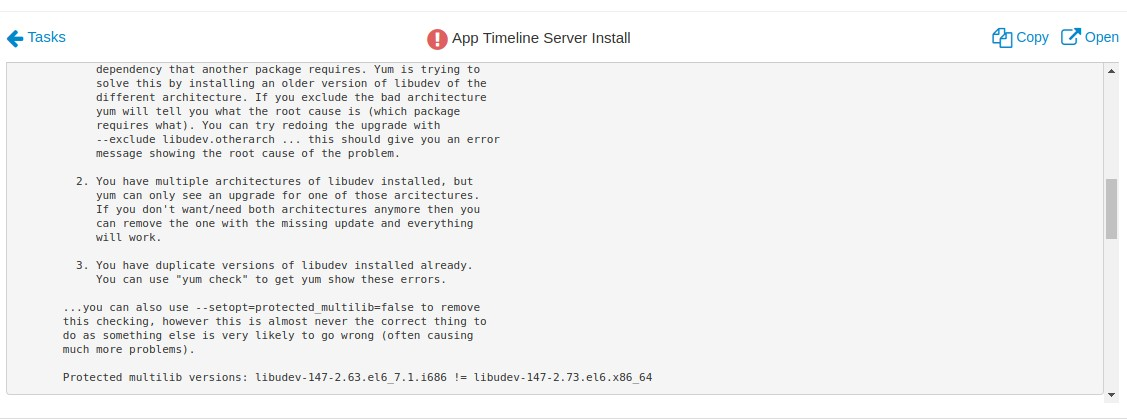
* Hive依赖MYSQL作为元数据库，装hive之前先装MYSQL
* MYSQL服务不能安装在ambari-server所在的机器，因为ambari-server机器已有mysql服务。
* 注意BD-OS或LuBan的所有服务需要在MySQL INIT以后，才能正常的启动和提供功能服务，并且MYSQL INIT之后，需要重启所有BD-OS或者LuBan服务
* 当重启hdfs时，会影响到依赖他的hive、hbase、HAWQ等服务，如有必要可以依次重启一下。
* 开启权限之前，请确保BD-OS或LuBan服务组件都已经安装好，并且license已经授权可以正常登录BD-OS或LuBan
* 除BD-OS外的第三方数据源如Oracle、Mysql、SQLServer、DB2等服务器时间区一定要和BD-OS服务器所在时区一致，比如都是CST
* spark2 thrift server必须至少有一个是和bdos runner server安装在同一台主机。
* storm的supervisor服务必须有一个是和bdos runner server安装在同一台主机。
* 如果只使用安装LuBan服务，那么不需要安装HAWQ和KHAN、HBASE，BD-OS STANDARD也不需要安装，只需要BASE SERVICE和 BDOS RUNNER
* 应用模块是通过bdos-license授权，ambari上的服务安装也是根据前期需求提前规划好，人为管控的，并不与mysql组件安装的4个配置产生强联系，不要混淆
* Hawq开启kerberos后，必须执行Run Service Check按钮对hawq服务是否可用进行检查

### 7.3.1安装中的常见问题

#### 7.3.1.1系统依赖包冲突问题

如下图的libudev版本冲突

注意：此问题是centos6中遇到的，centos7上处理思路相同

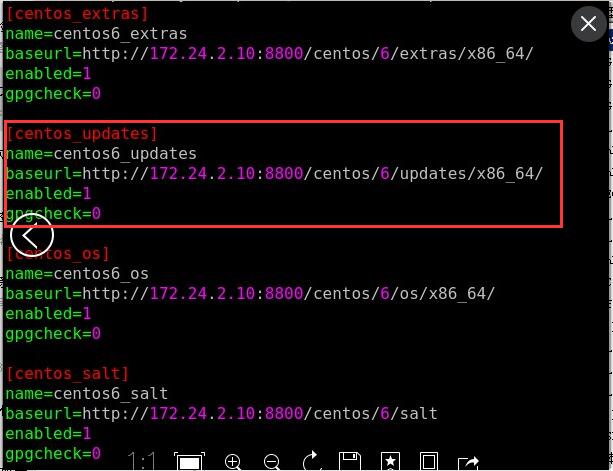


**图 版本冲突**

解决办法(示例)：从本地yum源获得错误提示需要的版本包（这里也可以从互联网查找获得），然后安装到本地，操作如下:

查看本地yum源文件

#> cat /etc/yum.repos/local.repo



如上图，优先找centos\_updates的源地址

浏览器输入yum源地址，如http://hostip:port/centos/6/updates/x86\_64/



**图 yum源地址**

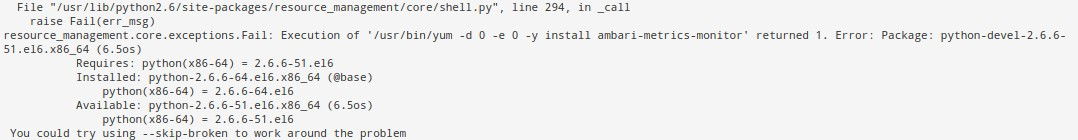
搜索libudev包下载到本地，上传到安装服务器（多台），

使用命令 rpm -ivh libudev-\* --force --nodeps 安装；

安装完成之后点击retry按钮，重试安装。

遇到其他包冲突问题大部分可以按照此种办法处理解决。

注意：尽量不要卸载本地系统已经安装的报冲突的包，容易导致系统错误无法继续安装。



如上图，同样也是包冲突问题，系统已经安装python-2.6.6-64，但是yum源要安装的python-devel的版本却是2.6.6-51，这个版本依赖的是python-2.6.6-51，环境多少会有这种奇葩的问题

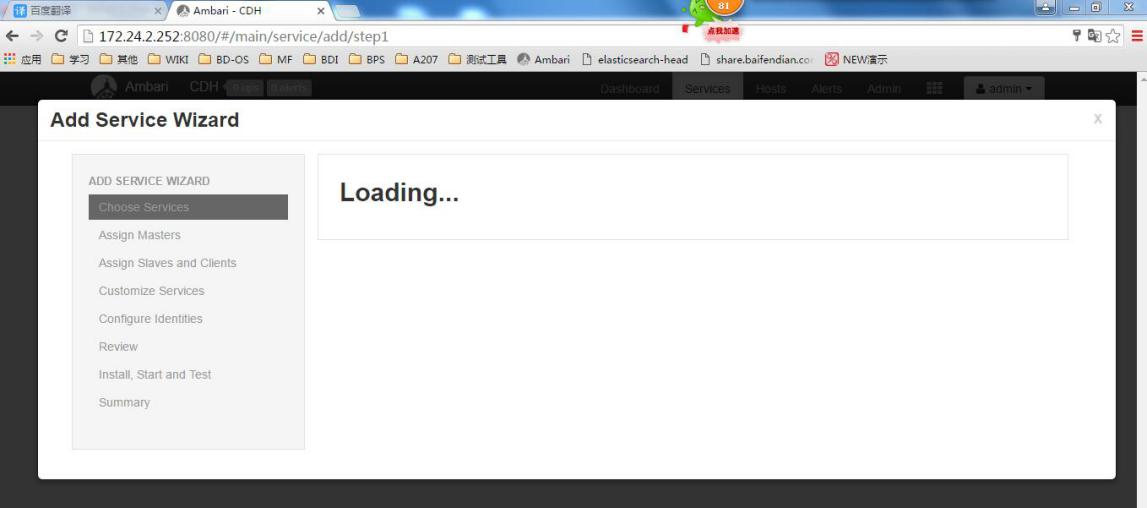
解决思路是：我们在本地的iso镜像里或者公网上找到python-devel-2.6.6-64这个rpm包，上传到集群服务器上，所有主机执行如下命令安装：

rpm -ivh python-devel-2.6.6-64.el6.x86\_64.rpm

这样在重试（try）安装，ambari查找包已经安装就会跳过顺利进行下面的安装。根据这种解决思路，遇到其它包冲突问题我想也不难解决

#### 7.3.1.2安装服务页面空白

当在添加服务时页面出现如下情况：



**图 添加服务**

进入终端后执行mysql进入mysql控制台执行如下操作：

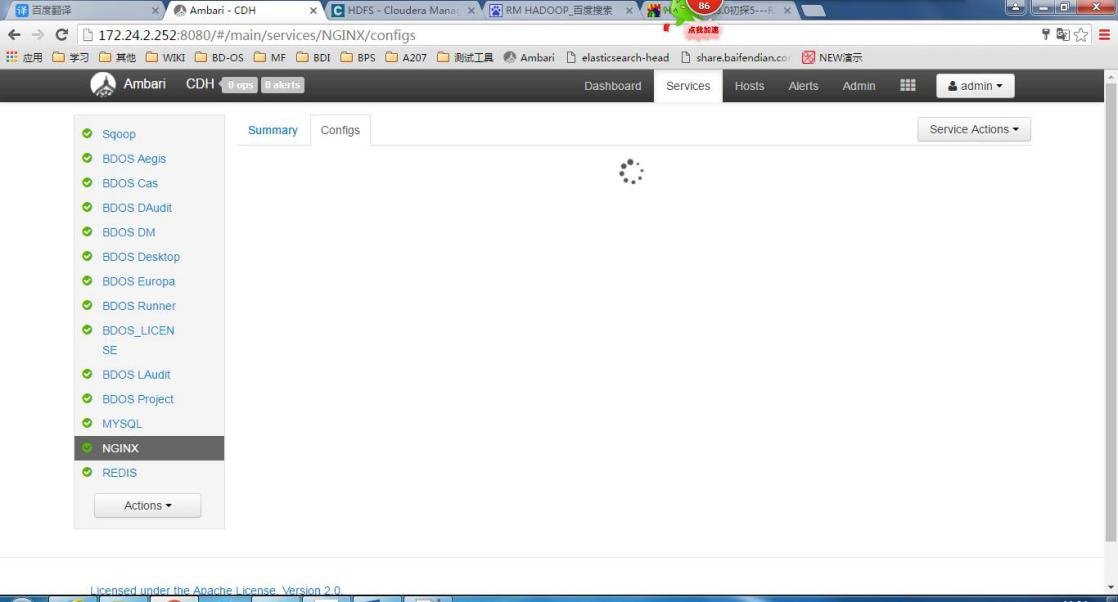
mysql> UPDATE ambari.cluster\_version SET state ='CURRENT';

然后重启ambari：

# ambari-server restart

#### 7.3.1.3服务Config界面空白

如果出现以下界面：

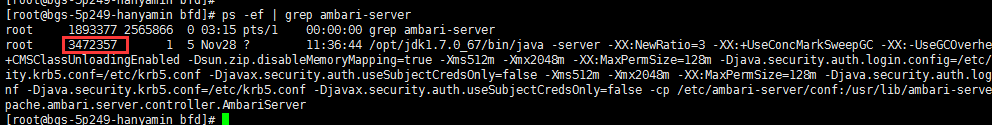


**图 服务Config界面空白**

杀掉ambari-server进程后重启ambari-server:

执行

# ps -ef | grep ambari-server



图中标记位置为ambari-server的进程号。

#kill -9 进程号

#ambari-server restart

#### 7.3.1.4域名中途变更

后期如果有修改域名的需求。

需要在nginx配置中修改域名并保存，然后需要所有bdos客户端机器上修改hosts文件，修改或添加正确的ip和域名对应关系。

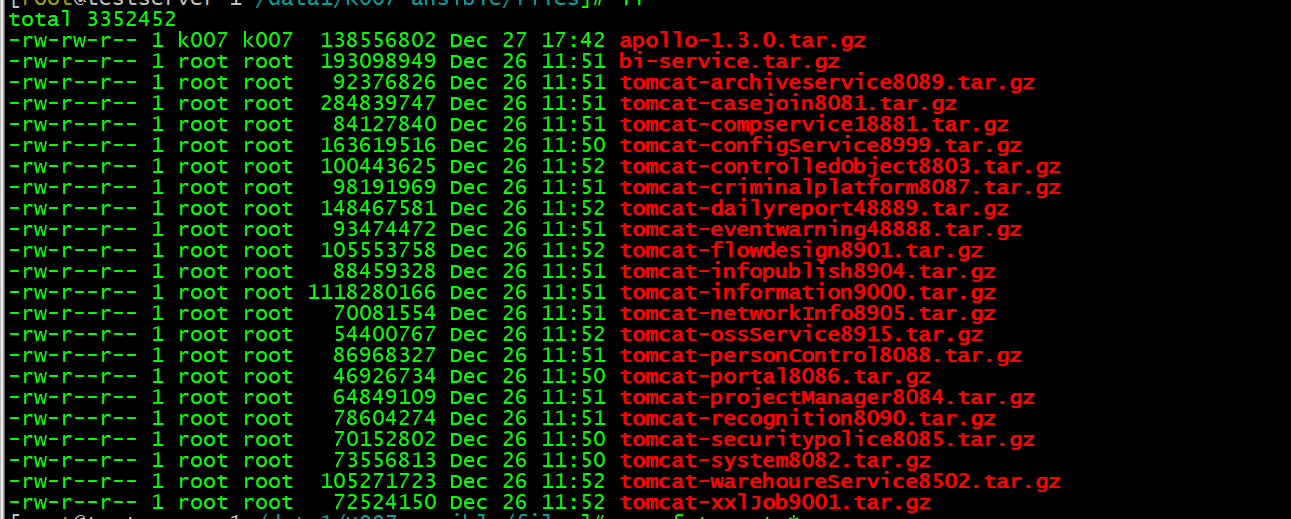
对Mysql进行重新重新初始化（initialize），将新域名写入配置表中。

然后重启Nginx服务和bdos、luban所有服务。

# 8 业务系统tomcat部署

Tomcat部署前要有以下准备：

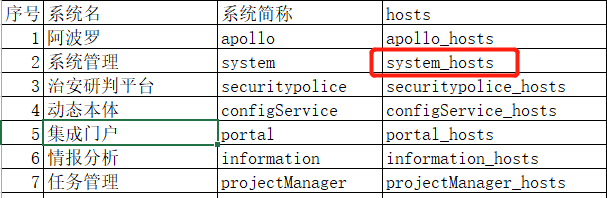
1. 保证Apollo地址可用
2. ansible脚本中关于tomcat的变量修改正确
3. 数据库中数据可用
4. 业务系统后端部署包都放在/etc/ansible/files下

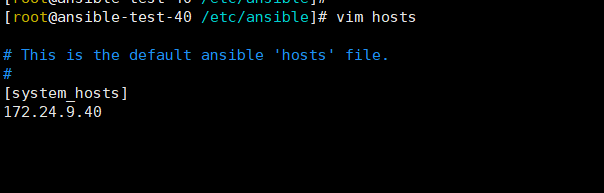


## 8.1单个系统部署流程(以部署系统管理为例)

### 8.1.1修改hosts文件

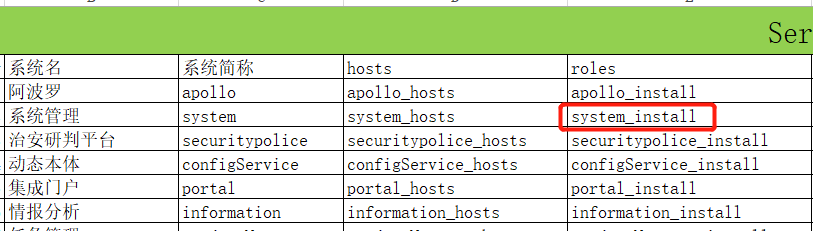
在/etc/ansible/hosts文件中添加对应系统组，系统组的名称参考ServiceList.xlsx，系统组下面添加要部署机器的ip地址。

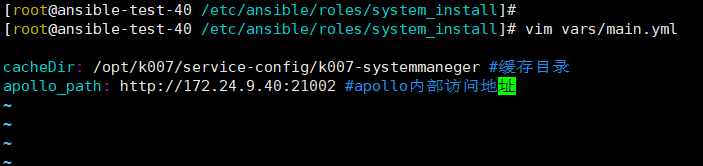




### 8.1.2修改参数

根据ServiceList.xlsx找到对应系统的roles，修改vars/main.yml,按照参数列表，修改对应参数。





### 8.1.3启动

命令：ansible-playbook /etc/ansible/playbooks/系统简称开头的yml文件。

ansible-playbook /etc/ansible/playbooks/system.yml

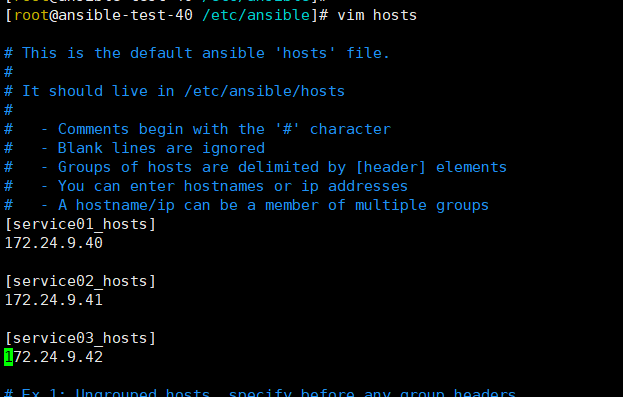
## 8.2多系统部署流程

按照如下结构部署为例

|  |  |  |  |
| --- | --- | --- | --- |
|  | ip | 系统名称 | 简称 |
| service01 | 172.24.9.40 | 阿波罗 | apollo |
| 系统管理 | system |
| 治安研判平台 | securitypolice |
| 动态本体 | configService |
| 集成门户 | portal |
| 情报分析 | information |
| 任务管理 | projectManager |
| 专题分析 | bi |
| service02 | 172.24.9.41 | 犯罪研判 | criminalplatform |
| 比对分析 | compservice |
| 情报发布 | infopublish |
| 情报识别 | recognition |
| 网络情报分析 | networkInfo |
| 重大事件预警 | eventwarning |
| 重点人员管控 | personControl |
| 串并案 | casejoin |
| service03 | 172.24.9.42 | 电子档案 | archiveservice |
| 流程设计中心 | flowdesign |
| 对媒体服务 | ossService |
| 受控对象 | controlledObject |
| 安全日报服务 | dailyreport |
| 业务系统-调度服务 | xxlJob |
| 数仓接口服务 | warehoureService |

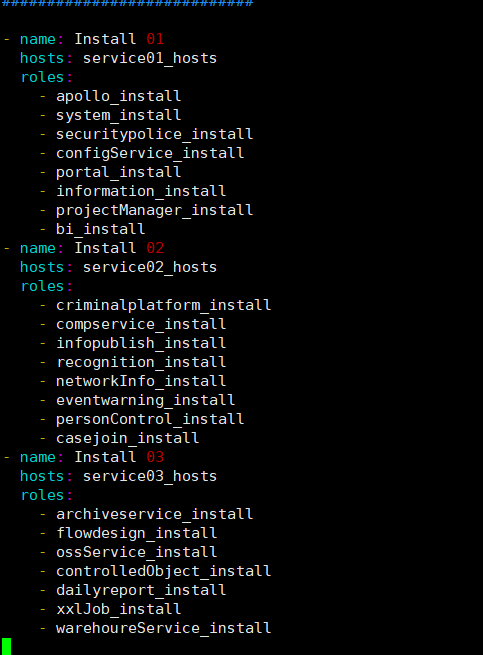
### 8.2.1修改hosts

编辑/etc/ansible/hosts文件，添加组名。



### 8.2.2修改对应playbook

编辑/etc/ansible/playbooks/service\_all.yml，按照如图结构修改。



### 8.2.3修改单个系统参数

启动前先修改每个系统对应的vars/main.yml里面的参数。

### 8.2.4启动

Ansible-playbook /etc/ansible/playbook/service\_all.yml

# 9 业务系统web服务部署

ansible-playbook /etc/ansible/playbooks/web\_update.yml

# 10其他服务部署

### 安装kafka

编辑 /etc/ansible/hosts 文件修改 [kafka\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/playbooks/kafka.yml

验证安装成功

ansible kafka\_hosts -m wait\_for -a "port=6667 delay=0 timeout=2"

### 安装sftp

编辑 /etc/ansible/hosts 文件修改 [sftp\_hosts] 标签下的内容，然后执行如下

ansible-playbook /etc/ansible/playbooks/sftp.yml

验证sftp是否启动

ansible sftp\_hosts -m service -a "name=sshd state=started"